

[Anasayfa](#) / [Basit Dili Kuralları](#)

Prosedürler

Prosedürler bir yemek-tarifi kitabının sayfaları gibidir. Her sayfa bir yemeğin tarifini yapar. Mesela 17ci sayfadaki yemek tarifine bakalım.

Adım adım yemeğin nasıl hazırlanacağı hakkında bilgi verir.

Bir prosedürde aynen böyle belli bir işi halletmek için tüm gereken adımları tek komut altında toplar.

Bu belli işi yapmak için her zaman o işi gören prosedürü çalıştırmak yeterlidir. Basitte tüm kodlamalar prosedürler ile yapılır. Prosedürlerin kapsama alanları “lokal birim” olarak belirlenir.

Prosedürlerin özellikleri:

- Parametre girilebilir
- Parametrelere standart değer verilebilir
- Bir sonuç geri verebilir
- Yükleme yapabilir

Prosedür Tanımı

En basit tanımlama yöntemi:

```
Prosedür  ADI  (  
    ...  
ProsedürSon
```

Sonuç geri vermeli:

```
Prosedür  ADI  (  
    ...  
ProsedürSon
```

Parametrelili:

```
Prosedür  ADI  (  Rakam  rak1;  rak2;  Metin  met  )  
    ...  
ProsedürSon
```

Standart değerli parametrelili:

```
Prosedür  ADI  (  Rakam  rak1;  rak2  =  5;  Metin  met1  =  "A";  met2;  Ba  
yt  c  =  0  )  
    ...  
ProsedürSon
```

Parametrelili ve sonuç geri vermeli:

```
Prosedür ADI ( Rakam rak1; rak2; Metin met ) Dizilimsel Virgüldür
...
ProsedürSon
```

Prosedürü kodlamak yeterli değil. İşlem yapması için, onu işlem yapacağı kod bölümünde çalıştırmak gerekir. Prosedürleri hemen tanırırsın. Yapı her zaman aynıdır. Prosedürün adı gelir ve hemen ardına () çifti gelir. Eğer parametreleri yoksa parantez açılır ve kapanır. Eğer parametreleri varsa, o halde parantez için parametreler girilir. Birden fazla parametreyi ayırt etmek için Semikolon işareti kullanılır:

```
Prosedür deBirŞey( Metin neDiyim; başkaNeDiyim )
    MetinselDenetim neDiyim && başkaNeDiyim
ProsedürSon

Prosedür Başlat ()
    deBirŞey( "Merhaba"; "Dünya" )
ProsedürSon
```

Prosedürün parametre yapısı aynı değişken ve dizilim tanımlama gibidir.

Örnek:

```
Prosedür ADI ( Rakam a; b; Virgül c; d; Metin e; f ) ...
```

Prosedür işlem yaptıktan sonra geriye bir sonuç verebilir. Eğer bir sonuç geriye verilecekse, bunu belirtmek lazım. Örnek:

```
Prosedür deBirŞey() Metindir
    Sonuç "Merhaba Dünya"
ProsedürSon

Prosedür Başlat ()
    MetinselDenetim deBirŞey()
ProsedürSon
```

Sonuç verme türleri normal, dizilimsel, işaretli, dizilimsel işaretli veya işaret olabilir:

Basitin Sonuç Geri Verme Türleri

Basit profesyonel bir yazılım dili olarak bazı temel veri türlerini sunar ve bunun dışında yeni veri türleri oluşturabilir. Bu bölümde temel veri türlerini tanıyacağız:

Veri Türü	BitAdeti	Minimum	Maksimum	Amacı
YarıBayttır	8	-128	+127	En düşük sayısal değerleri karşılamak içindir

Veri Türü	BitAdeti	Minimum	Maksimum	Amacı
Bayttır	8	0	255	En düşük, fakat pozitif olan sayısal değerleri karşılamak içindir
Asciidir	8	0	255	Ascii Tablosunun karakterlerini desteklemek içindir (Özellikle C fonksiyonlarıyla veri alışverişinde bulunurken gereklidir)
ÇiftBayttır	16	0	65535	Geniş bir banda sahip olup, sadece pozitif sayısal değerleri karşılamak içindir
SafKüçükRakamdır	16	0	65535	Geniş bir banda sahip olup, sadece pozitif sayısal değerleri karşılamak içindir
KüçükRakamdır	16	-32768	+32767	Geniş bir banda sahip olup, hem pozitif, hemde negatif sayısal değerleri karşılamak içindir
Renktir	32	0	4294967295	KYM (RGB) ve KYMŞ (RGBA) renklerini karşılamak içindir
Saattir	32	0	4294967295	Saat, Dakika ve Saniyeyi karşılamak içindir
Rakamdır	32	-2147483648	+2147483647	Standart sayısal değerleri karşılamak içindir
SafRakamdır	32	0	4294967295	Standart geniş bir banda sahip olup, sadece pozitif sayısal değerleri karşılamak içindir
EsnekRakamdır	32	-2147483648	+2147483647	32 Bit derlemede Rakam ile aynıdır
SafEsnekRakamdır	32	0	4294967295	32 Bit derlemede SafRakam ile aynıdır
EsnekRakamdır	64	-9223372036854775808	+9223372036854775807	64 Bit derlemede GenişRakam ile aynıdır
SafEsnekRakamdır	64	0	18446744073709551615	64 Bit derlemede SafGenişRakam ile aynıdır

Veri Türü	BitAdeti	Minimum	Maksimum	Amacı
GenişRakamdır	64	-9223372036854775808	+9223372036854775807	Çok geniş bir banda sahip olup, hem pozitif, hemde negatif sayısal değerleri karşılamak içindir
SafGenişRakamdır	64	0	18446744073709551615	Çok geniş bir banda sahip olup, sadece pozitif sayısal değerleri karşılamak içindir
KüçükVirgüldür	32	-1175494e-38	+3402823e+38	Normal boyuttaki virgüllü rakamlar içindir
Virgüldür	64	-22250738585072013e-308	+17976931348623157e+308	Büyük virgüllü rakamlar içindir
İzdir	32	0	4294967295	Türü belirsiz, işaretli adresleme içindir
Tarihtir	32	-2147483648	+2147483647	Sadece Tarih bilgisini içermek içindir
Tarihtir	64	-9223372036854775808	+9223372036854775807	Sadece Tarih bilgisini içermek içindir
TarihSaattir	32	-2147483648	+2147483647	Tarih ve Saat bilgisini içermek içindir
TarihSaattir	64	-9223372036854775808	+9223372036854775807	Tarih ve Saat bilgisini içermek içindir
AsciiMetnidir	*	Bayt		Bu tür C kütüphaneleri ile veri alışverişi için özel olarak sunulur. Basit standart olarak Metin türünü kullanır ve tüm karakterleri Unikod modelinde Metin türünde tutar
Metindir	*	ÇiftBayt		Basitin sunduğu standart karakter türüdür. Tüm metinsel veriler bu türle tutulur

Veri Türü	BitAdeti	Minimum	Maksimum	Amacı
Esnektir	32	0	4294967295	Esnek özel bir türdür. Tüm diğer türleri destekler. Bazen gerekli olduğu için sunulmaktadır. Fakat standart olarak asla kullanılmamalıdır. Sadece gerçekten gerektiği yerde kullanılması doğrudur.
Esnektir	64	0	18446744073709551615	Veri Türü "Esnek" özel bir türdür. Bu türün özelliği ise, diğer türlerin hepsinden olabilir. Esnek ihtiyaç durumunda yeniden yapılandırılır. Esnek türünün kullanılmasının gerekli olduğu durumlar mutlaka vardır. Ama önerilen o ki, mümkün oldukça bu türden az kullanılmasıdır. Çünkü "Esnek" normalden çok daha fazla hesaplama ve işlem adımları gerektiriyor. Bu durum programın yavaşlamasına ve ağır çalışmasına sebebiyet verebilir. Tabloya bakıldığında kapasite açısından aynı kapasiteye sahip türler ile karşılaşıyoruz. Kapasiteleri aynı olsa bile kullanım durumlarında farklılıkları ortaya çıkmaktadır.
Değişkendir	32	0	4294967295	Sayısal veya Virgöl geri verir

Veri Türü	BitAdeti	Minimum	Maksimum	Amacı
Dizilimdir	32	0	4294967295	Dizilim geri verir
Yapıdır	32	0	4294967295	Yapısal adres geri verir
Objedir	32	0	4294967295	Objesel adres geri verir
Bileşendir	32	0	4294967295	Bileşen adresi geri verir
İşarettir	32	0	4294967295	Bellek adresi ve salt kaynak adresi geri verir
İşarettir	64	0	18446744073709551615	Bellek adresi ve salt kaynak adresi geri verir

Örnekler:

Prosedür X (...) Rakamdır ? Normal bir sayısal değer verir
 Prosedür X (...) Dizilimsel Rakamdır ? Komple bir dizilimi geri verir
 Prosedür X (...) İşaretli Rakamdır ? Global birimden bir değışkene işaret verebilir
 Prosedür X (...) İşaretli Dizilimsel Rakamdır ? Global birimden bir dizilime işaret verebilir
 Prosedür X (...) İşarettir ? Bellek alanında bir adrese işaret verebilir
 Prosedür X (...) Yapıdır ? Bir yapı geri verebilir
 Prosedür X (...) Objedir ? Bir obje geri verebilir

Erişebilirlik

Prosedürlerin tanımlama sırası önemli değildir. Basit derleme esnasında gereken tüm önemlemleri otomatik olarak alır. Aynı geçerlilik alanında bulunan tüm prosedürler birbirleri tarafından çalıştırılabilir. Tek istisna Başlat-Prosedürüdür.

From:

<https://www.basit.web.tr/dokuwiki/> - Basit Yazılım Dili Dokümantasyonu

Permanent link:

<https://www.basit.web.tr/dokuwiki/doku.php?id=proseduerler>

Last update: **11.10.2020 21:51**

